

Інноваційні підходи до створення колективної пам'яті на основі парадигми Semantic Web

О.В. Новицький

Abstract – В роботі представлено ряд теоретичних ідей які можуть бути втілені при створенні електронної бібліотеки (ЕБ). Зокрема значна увага приділена яким чином технологія Semantic Web використовується в різних аспектах ЕБ. Виділено основні рівні в структурі ЕБ які можуть бути носіями семантичного описання. Описані переваги такого підходу.

Keywords – електронні бібліотеки, семантичний веб, колективна пам'ять.

I. ВСТУП

Електронна бібліотека є складною інформаційною структурою. Саме поняття електронна бібліотека на даний момент конкретно не визначено. В роботах [1]-[4] було проведено аналіз стосовно цього поняття. Ми зупинимося на такому підході, що електронна бібліотека це об'єднання через мережу електронних текстів, документів, зображень, звуків, наукових даних та програмного забезпечення яке є ядром сьогоденного Інтернету, а в майбутньому через організацію доступу до електронних бібліотек буде утворюватися база знань людства. Цей підхід породжує так звану колективну пам'ять. Поняття колективної пам'яті саме для цифрових бібліотек виникло відносно недавно. Проте цей термін набув широкого поширення, зокрема комітет IEEE Technical Committee on Digital Libraries, трактує це поняття як сукупність електронних бібліотек, електронних музеїв, електронних архівів. Зазвичай основна інформація яка передавалася це була текстова інформація, про те на разі, передається інформації інших типів відео, звук, фотографії та ін..[5].

Такий підхід є інноваційним оскільки дає цілісний доступ до інформації будь-яким користувачам будь-де. Розвиток сучасних інформаційних технологій дає можливість реалізувати це.

Оскільки сучасний науковий пошук пов'язаний з великою кількістю даних, тому важливо щоб ця інформацію могли використовувати всі.

Цей підхід дає можливість безпосередньо просліджувати виникнення наукових даних. Тим самим можна гарантувати їх достовірність, водночас зберігати унікальні екземпляри в бібліотеках, музеях та архівах. Сьогодні в багатьох музеях та бібліотеках зберігається величезна кількість безцінної інформації, проблема в тому, що до неї є поки тільки фізичний доступ. Створення колективної пам'яті позитивно відобразиться на науці, такий підхід повинен стати серйозним поштовхом.

О.В. Новицький – Інститут програмних систем НАН України, проспект Академіка Глушкова 40, 03187, Київ, Україна. alex@zu.edu.ua

Відомо, що кожна наукова група збирає та поповнює свій інформаційний фонд, причому якщо кожна така група працює в одному напрямі то відповідно і інформаційні фонди будуть одного змісту, але можливо різної структури, що призводить до неефективної роботи, та різного тлумачення наукових понять.

Обмін інформацією дасть можливість аналізувати дані спостережень одночасно багатьом науковим групам навіть якщо вони будуть знаходитися на дуже великих відстанях. Таким чином утворюється спільний робочий простір, що дозволяє всім взаємовигідно працювати над проблемою.

Колективна пам'ять утворює так звані портали знань та контенту, що представляє собою мережу з розподіленими ресурсами.

Розвиток колективної пам'яті одночасно потребує розвитку інших напрямів.

- **Зберігання.** Система Колективної пам'яті повинна та здатна зберігати великі об'єми інформації різномірних форматів. Збереження текстової інформації самий простий випадок. Проте мультимедійну інформацію, зображення, зберігати набагато складніше оскільки цей тип інформації потребує великого об'єму пам'яті та потрібен певний час для доступу до такої інформації. Окрім того, вона погано піддається формальній обробці.

- **Інтерфейс користувача.** Один з самих важливих компонентів колективної пам'яті, який повинен представляти велику кількість сервісів, для взаємодії між користувачем та інформацією яку він шукає. В доповнення для звичайних функціональних можливостей пошуку, інтерфейс користувача колективної пам'яті повинен забезпечувати нові можливості, наприклад обробку відео, анотації та ін..

- **Класифікація та індексація.** Дає змогу групувати об'єкти. Однак виявлено, що на це сильно впливає індивідуальне сприйняття та великий обсяг інформації яку необхідно індексувати. Автоматизовані методи класифікації поділяють на статистичний та аналітичний. Підходи до класифікації зображень сильно відрізняються від звичайних текстових. Відео класифікація потребує системи, які зможуть аналізувати з точки зору семантичної релевантності. Класифікація аудіо даних потребує додаткового дослідження

- **Інформаційний пошук.** В цій області існує багато методів пошуку, включаючи пошук мета даних та контенту. Визначити корисність результату пошуку може тільки сам користувач. Для покращення ефективності використовують додаткові метадані, які описують документ. Дослідники також зосереджуються на автоматизації створення і обслуговування параметрів користувача для використання їх в процесі пошуку. Використання програмних агентів дозволить покращити

- Представлення. Користувачі Колективної пам'яті можуть надавати різну перевагу до вигляду представленої інформації. Системи колективної пам'яті повинні бути дуже гнучкі відповідно до переваг кожного користувача. Вони повинні знати про можливості апаратних засобів та її обмеження, автоматично коректуючи для максимальної якості.

Окрім розглянутих вище напрямків необхідно також збільшувати ступінь деталізації. Створення нових схем мета даних, зосередження на інформаційному вмісту а не на інформаційних об'єктах. [6].

Отже оперування такими обсягами інформації породжує певні проблеми. З погляду на вище сказане ці проблеми можна розділити на дві великі групи, перша група проблем пов'язана з технічними труднощами організації збереження інформації та доступу до неї, друга група проблем пов'язана з логічною організацією колективної пам'яті та забезпечення доступу до неї з подальшим аналізом змісту. Ця робота стосується вирішення проблем 2-гої групи.

Для того щоб показати переваги семантичного підходу до електронної бібліотеки на відміну від класичної необхідно виділити ті структурні елементи які раніше не мали семантичної моделі, і до яких ми пропонуємо цю модель побудувати.

Основними двома компонентами електронної бібліотеки є її *контент* та *набір програмного забезпечення* для роботи з цим контентом. Для початку розглянемо контент ЕБ. Інформація в електронних бібліотеках описується в термінах *електронні об'єкти* (*Digital objects - DO*), які являють собою мультимедійний контент і метадані [8]. Оскільки обсяги DO значні, то для спрощення пошуку та класифікації використовують анотування DO.

Формальна модель, яка запропонована в [10], виділяє два підходи до розуміння анотацій: анотації як метадані або анотації як контент.

У першому випадку ми маємо справу з різноманітними схемами метаданих (Dublin Core, MARC і ін.) які використовуються для опису інформаційних ресурсів. Ці анотації насамперед направлені на користувача.

У другому випадку анотації як надання контенту призначені для автоматизованої машинної обробки. Ці анотації надають *семантику* документа. Семантична анотація - анотація написана формальною мовою з добре визначеною семантикою і заснована на онтологіях. Фактично ці анотації є формальною моделлю DO, з можливістю машинної обробки.

Семантика контенту в свою чергу, може визначатися на основі зовнішніх зв'язаних онтологій, що дозволить будувати семантичну модель документу, де зв'язок визначається між окремими сегментами DO, та на основі семантики зв'язків між структурними компонентами DO, де зв'язок визначається між логічними закінченими структурними компонентами Рис. 1.

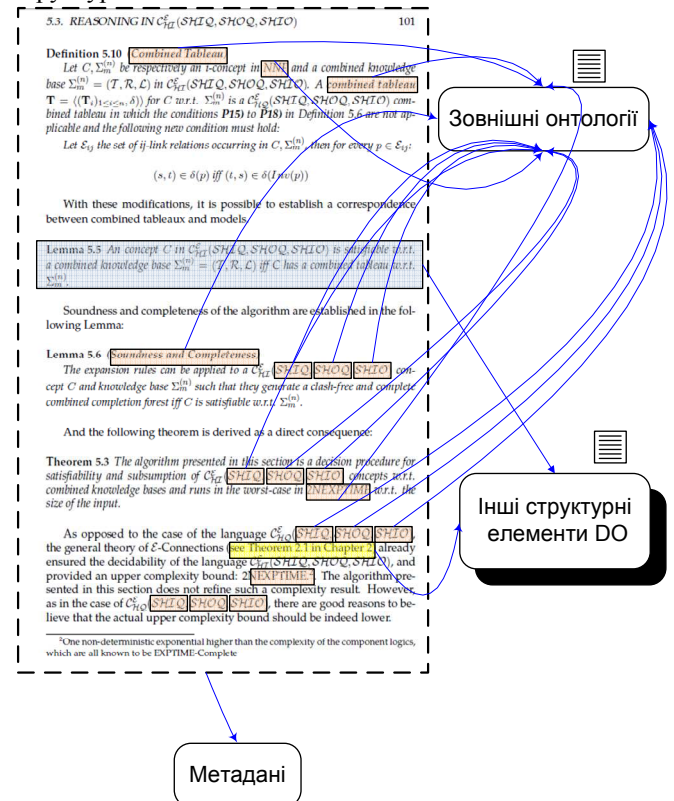


Рис.1 Анотування документа шляхом визначення відношень між термінами, лемами та зовнішніми отологіями а також визначення зв'язків між логічно закінченими елементами з іншими частинами DO. Окрім цього DO описується метаданими.

Модель, яка представлятиме цифровий об'єкт повинна відображувати фактичний зміст даного об'єкту. Серед безлічі розроблених моделей, ми використовуємо модель, яка запропонована в [8], [11] зі змінами та уточненнями, які враховують раніше описану модель.

У кожен момент часу k ми маємо деяку множну цифрових об'єктів $DO(k)$, при анутованні документа, ми отримуємо деяку нову множину документів у момент часу $DO(k+1)$. Далі якщо деяка змінна явно залежатиме від k то на цьому буде наголошуватиметься, інакше змінна від k незалежна.

Розташування кожного цифрового об'єкту також як і анотації ідентифікується унікальним ідентифікатором –

посиланням (link). Окрім цього посилання сполучає цифровий об'єкт і анотацію, і може відображувати відношення між об'єктами. Отже, можна виділити два типи посилань:

посилання анотації (Annotate link) - відображує відношення в середині цифрового об'єкту, який може бути як документом, так і анотацією;

посилання відношення (Relate-to link) - визначає відношення зовнішнього цифрового об'єкту до об'єкту, що анотується.

Нехай LT множина типів посилань, тоді LT містить наступні типи посилань $LT = \{AnnotateLink, Relate - to Link\}$.

Посилання надаються у вигляді ідентифікаторів. Загальноприйнятими ідентифікаторами можуть виступати URI, DOI, OPENURL, Persistent URL (PURL), PURL-based Object Identifier. $H(k)$ множина ідентифікаторів цифрових об'єктів у момент часу k .

У цій моделі цифровий об'єкт надається у вигляді потоку. Потік sm це кінцева послідовність:

$$DO(\Sigma) \rightarrow sm : I = \{1, 2, \dots, n\}, n \in \mathbb{N}$$

де $\Sigma = (\varphi_1, \varphi_2, \dots, \varphi_n)$ - алфавіт символів.

Якщо ми маємо потік sm :

$$DO(\Sigma) \rightarrow sm : I = \{1, 2, \dots, n\}, n \in \mathbb{N},$$

то в цьому потоці ми можемо виділити неперервний сегмент st_{sm} послідовності чисел a, b так що:

$$st_{sm} = [a, b], 1 \leq a \leq b \leq n, n \in \mathbb{N}.$$

Безліч сегментів ми позначаємо ST , так що $\forall st_{sm_i} \in ST, i = 1, 2, \dots, n$.

Якщо цифровий об'єкт DO має безліч унікальних ідентифікаторів H то функція hsm відображує унікальний ідентифікатор до сегментів які містяться в DO :

$$h \xrightarrow{hsm} st_{sm_i}, n \in \mathbb{N}$$

Потік вимагає, щоб функція не мала властивостей сюр'єктивності і інєктивності. Кожен цифровий об'єкт може мати принаймні один потік.

SM визначає множину потоків, так що $\forall sm_i \in SM, i = 1, 2, \dots, n$.

Анотація, яка додається до цифрового об'єкту, може бути різного типу. SN - визначає множину типів анотацій, яка додається до цифрового об'єкту (текстова анотація, відео анотації, анотація у вигляді малюнка, анотація у вигляді звуку). Ця анотація теж представляється у вигляді потоку, тобто $SN \subseteq SM$. Одиначний вид анотації позначатимемо sn , так що $\forall sn \in SN$.

Анотація $a \in A(k)$ це кортеж:

$$a = \left(h_a \in H(k), A_\alpha \subseteq SN(k) \times LT \times ST(k) \times \times SM(k-1) \times H(k-1) \right)$$

де h_a - унікальний власний ідентифікатор анотації a , тобто $h(h_a) = a$;

A_α множина n -арних відношень анотації a і визначається як добуток множин SN, LT, ST, SM та H .

У випадку анотування веб-документів, формальна модель зміниться. Нехай A множина всіх анотацій a , а D множина документів, відповідно, $DO = D \cup A$, причому підмножина множини DO позначатимемо do , тобто $do \in DO$.

Анотацією веб-документа називатимемо мічений граф:

$$G := ((DO, E_{da} \subseteq A \times DO))$$

Де $DO = D \cup A$ вершини графа;

$$E_{da} = \left\{ (a, do) \in A \times DO \mid \exists \alpha \in A_\alpha, \alpha = (sn, sm_i, st_{sm_i}, hsm^{-1}, LT) \right\} - \text{сторони графа}$$

Розглянемо архітектуру програмного забезпечення. Ми вважаємо, що використання сервіс-орієнтованої архітектури є ключовим елементом досягнення інтероперабельності. У такому середовищі складне програмне забезпечення може бути спроектоване на основі сервісів, що є у розпорядженні і які доступні не лише локально, але і через Інтернет. У такому контексті для побудови електронних бібліотек ми пропонуємо використовувати архітектуру сервіс-орієнтованої електронної бібліотеки (Service-Oriented Digital Library architecture - SODL) [7], це забезпечить зручний спосіб для досягнення побудови колективної пам'яті.

Для забезпечення вільного доступу служб до бібліотеки вона має бути реалізована на відкритій архітектурі. Тобто взаємодія служб має бути описана за допомогою стандартизованих правил і середовища, що дасть можливість вільно вводити новий сервісний елемент, не перебудовуючи систему наново.

Для того, щоб забезпечити динамічне налаштування електронної бібліотеки, необхідно забезпечити механізми, які виконуватимуть цю функцію. Таким механізмом є введення семантики в середовище функціонування веб-сервісів. Цей підхід отримав назву Semantic Web, і стосується не тільки семантичного опису сервісів а також і семантичного опису інформаційних ресурсів Інтернету (веб-сторінок, документів тощо.)

Під *сервісом* ми розуміємо деякою послугу мета якої задовольнити запити користувача. Сервіси реалізуються за допомогою веб-сервісів. Веб-сервіс це програмне застосування. Основна відмінність між сервісом і веб-сервісом полягає в тому, що той самий сервіс одночасно можна реалізувати різними веб-сервісами. У свою чергу

веб-сервиси можуть між собою об'єднуватися, створюючи новий веб-сервіс, цей процес називають *композиція* веб-сервісів.

Сервиси електронної бібліотеки можна класифікувати за різними аспектами. Зокрема результатом виконання деякого сервісу на вимогу запитуючої сторони буде результат у виді структурованої інформації про електронні об'єкти які містяться в ЕБ. Тому якщо розглядати ЕБ як замкнене середовище, очевидно, що інформаційне наповнення ЕБ не зазнало змін, сервиси з такою властивістю будемо називати *сенсорні* сервиси.

Поняття сенсорних сервісів для ЕБ ми ввели тому, що ці сервиси аналогічно до сенсорів фільтрують інформаційне середовище, фактично не впливаючи на зміст цього середовища.

В рамках підходу Semantic Sensor Web [9] ми додатково ануємо результати виконання сервісів. Тому на відміну від класичної побудови семантичного середовища для веб-сервісів, де результат не анується, ми передбачаємо анування метаданими результатів виконання веб-сервісів. Характер метаданих повинен передбачати історію, спосіб, локалізацію та час отримання результатів. Для побудови електронної бібліотеки на основі сервіс-орієнтованого підходу необхідно виділити сервиси, на основі яких виконуватиметься композиція [12]. Це обумовлено тим, що завдання композиції простіше вирішувати і, крім того, вимоги до електронної бібліотеки є такими, що в першу чергу необхідно вирішувати композиційні завдання. При виділенні базових сервісів ми керувалися тим, що кожен базовий сервіс відповідає певному рівню абстракції структури бібліотеки. Базовими сенсорними сервісами електронної бібліотеки ми вважатимемо наступні:

Сервіс локалізації. Визначає місцезнаходження заданого об'єкту і шлях до нього.

Сервіс базового або простого пошуку. Забезпечує пошук по базових наборах метаданих.

Сервіс розширеного пошуку. Забезпечує пошук по всіх полях логічним комбінаціям полів метаданих.

Сервіс навігації по року видання. Займається побудовою дерева. Будує дерево на основі року видання.

Сервіс навігації по предметному покажчику. Будує дерево на основі предметного покажчика.

Сервіс навігації по структурних підрозділах. Будує дерево на основі структурних підрозділів.

Сервіс навігації по авторах. Будує дерево на основі авторів або творців об'єктів.

Всі наступні сервиси не відносяться до категорії сенсорних:

Сервіс перетворення форматів. Призначений для перетворення документів одного формату в інший. Цей сервіс є необхідним, оскільки він забезпечить роботу бібліотеки з різними форматами даних.

Сервіс індексу-цитовання. Забезпечує визначення рівня індексу цитування документа. Тобто на основі аналізу тексту, будується карта документів і об'єктів, на які посиляється вихідний документ.

Сервіс побудови графа цитування. Виконує функції побудови графа цитування, де вузлами графа будуть об'єкти або документи, а зв'язки між вузлами є власне посилання. При побудові семантичного графа вузлами можуть виступати, як ресурс або частка ресурсу, а також елементи бази знань, а зв'язки між вузлами виступають як стосунки між об'єктами.

Сервіс індексації. Відповідає за індексування ресурсів з урахуванням лексичних форм. Він також допускає вибіркоче індексування об'єктів.

Сервіс конвертації метаданих. Відповідає за конвертацію одних схем метаданих в інших.

Сервіс управління ресурсами. Забезпечить управління авторським правом, політиками і дублікатами об'єктів електронної бібліотеки.

Сервіс ідентифікації двійників ресурсів. Сервіс призначений для визначення схожості двох ресурсів. Це досить важливий сервіс, оскільки він виявляє перекриття між будь-якими частками ресурсу, тому дозволяє відшукувати дублікати на будь-якому рівні. Крім того, з використанням технології Semantic Web, чтане можливим визначати перекриття ресурсів певної наукової області.

Сервіс зберігання репозиторія. Займається зберіганням і резервним копіюванням.

Сервіс збереження мультимедійної інформації. Відповідає за збереження і завантаження потокової інформації.

Сервіс управління об'єктами. Відповідає за уявлення, видання і заміну документів.

Сервіс управління доступом. Забезпечує авторизацію користувачів, а також відновлення і встановлення паролів доступу. Також забезпечує багаторівневу систему доступу до інформації.

Сервіс управління користувачами. Призначений для управління правами групи користувачів системи.

Сервіс збереження результатів пошуку. Призначений для користувачів, які пройшли авторизацію, причому можливо вибіркоче збереження результатів.

Сервіс інтерфейсу користувача. Відповідає за взаємодію між прикладними сервісами і протоколами. Кожне таке обслуговування може бути централізованим або розподіленим.

Сервіс історії даних. Описує всі передумови і весь шлях виникнення і зміни даних.

Сервіс зворотного зв'язку. Забезпечує можливість зворотного зв'язку з автором об'єкту, на основі будь-якої доступної контактної інформації, яку вказав автор.

Сервіс анування інформації. Сервіс анування дозволяє анувати ресурси та формувати семантичні моделі DO. Ми вважаємо, що з огляду на розвиток технології Semantic Web в основному всі DO будуть мати свої семантичні моделі.

Сервіс інтеграції. Виконує роль побудови колективної пам'яті, шляхом об'єднання електронних бібліотек.

Нехай ми маємо запит r , у якого є початкові параметри r_{in} і бажані параметри виходу r_{out} , необхідно знайти веб-сервіс w такий, щоб виконував r причому виконувалися умови:

$$r_{in} \supseteq w_{in}$$

$$r_{out} \supseteq w_{out}$$

Проблема пошуку веб-сервісу, який може самостійно виконати запит r носить назву дослідження веб-сервісів (Web Service Discovery - WSD). Коли за допомогою одного веб-сервісу неможливо досягнути виконання запиту r , потрібно утворити композицію кількох веб-сервісів $\{w^1, w^2, w^3, \dots, w^n\}$ послідовним чи паралельним способом, таким чином, щоб для всіх $w^i \in \{w^1, w^2, w^3, \dots, w^n\}$, причому w_{in}^i може бути заснований на w_{out}^i , та виконувалось відношення $(r_{in} \cup w_{out}^1 \cup \dots \cup w_{out}^n) \supseteq r_{out}$. Ця проблема носить назву композиції веб-сервісів (Web Service Composition - WSC). Більш складною проблемою є проблема композиції веб-сервісів.

Цю проблему класифікують по наступних аспектах [13]. Ручний чи автоматичний спосіб композиції – цей аспект визначає методологію композиції веб-сервісів. При ручному способі людина яка як правило добре володіє знаннями про предметну область самостійно визначає які сервіси та в якій послідовності необхідно взяти для вирішення поставленої цілі. Однак цей процес є доволі складним малопритатним для масштабних проектів і незахищений від виникнення помилок викликаних людським фактором.

Автоматичний спосіб навпаки вирішує всі вище згадані проблеми, програма знає які веб-сервіси і яким чином можна між собою зв'язати для досягнення поставленої цілі.

Класифікація по простоті та складності операторів для композиції веб-сервісів. В самому простому випадку при композиції використовується тільки оператор AND. В більш складаному випадку при композиції використовуються оператори (OR, XOR, NOT) або міститься деякі обмеження при виборі веб-сервісів здійснюючи послідовну чи паралельну композицію.

Класифікація по масштабу застосування. В загальному випадку проблема WSC може бути зведена до проблеми виконання, яка більш відома як задача NP-complete.

Для велико-масштабних задач WSC алгоритми повного перебору не є ефективними, в даному випадку більш придатними є алгоритми автоматичного пошуку підоптимальних рішень вибірки сервісів на основі знань про предметну область. Введення семантичного опису веб-сервісів дасть можливість автоматично вирішувати задачі WSC та WSD.

Саме цей випадок ми будемо розглядати в даній статті.

Поряд з тим, в процесі функціонування сервіс-орієнтованої електронної бібліотеки середовище в якому функціонують веб-сервіси постійно змінюється. Зміна середовища породжується двома основними факторами:

- по-перше проблеми, які породжують розподілені системи часові затримки і ненадійність транспортного протоколу, недостача пам'яті спільного використання

між частинами розподіленої системи, проблеми відмови доступу та паралельних запитів, а також проблеми пов'язані з програмною несумісністю в наслідок оновлення частини розподіленої системи;

- по-друге людський фактор, в процесі обслуговування користувача, можуть змінитися вимоги користувача, а отже це певним чином впливатиме на результат. Пересічні користувачі цільової аудиторії не мають чіткого уявлення про архітектуру бібліотеки, а отже не можуть наперед чітко визначитися з цілями які повинні задовольняти сервіси.

Ми будемо розглядати композицію на основі ціле-орієнтованої парадигми, тобто виходячи початкових умов та наявної множини сервісів здійснити композицію. Причому, оскільки веб-сервіси розміщені в семантичному середовищі, то вибирати такі плани композиції, які можуть бути корисними для кінцевого користувача. Тобто на відміну від класичної постановки задачі, від специфікації цілі до пошуку сервісів, які зможуть цю ціль досягти, виходити з того припущення, що наявна множина сервісів може досягнути деякі наперед невідомі цілі, які можуть бути обрані користувачем.

В такій складній системі цілі, які постають перед композицією можуть змінюватися. Ми також виходимо з того, що знання про оточуючий світ є не повними а отже і цілі є неповними. Для таких конфліктних цілей Horst Rittel і Melvin Webber ввели поняття вікід (wicked) задачі. Тому класичні методи планування виявилися неефективними. В середовищі де цілі не є чітко визначеними і є динамічний процес планування приймає інше смислове значення, а саме, план виконання дій не є однозначним алгоритмом досягнення цілі, функція планів можна сформулювати наступним чином [14]:

- перевірка ресурсів;
- починають координаційні процеси або допомагають спростити координаційні процеси на початку;
- встановлюють відповідальність та ідентифікацію;
- трековий прогрес і що більш важливо генерація намірів;
- впізнають та управляють ризиками;
- підтримують імпровізацію, і що саме важливо формалізують представлення людини (користувача) про проблему яку план намагається вирішити.

Така комбінація функціональних властивостей призводить до того, що плани необхідно додатково досліджувати і проводити репланування. Тобто змінювати плани під час виконання цих планів.

Розглянемо найпростіший випадок в електронній бібліотеці. Користувачу необхідно знайти деяку інформацію, система генерує можливі варіанти пошуку інформації і користувач вирішив що краще для цього скористатися базовим пошуком, проте система не може однозначно визначити ціль (ціль неточна) а отже і можливий план розвитку оскільки в процесі виконання плану користувач може змінити ціль, а отже потрібно репланувати, тобто змінювати початковий план. При цьому користувач постає перед вікід-проблемою, оскільки

постає дилема вибору між важливістю можливо знайденої інформації та часом затраченим на її пошук Рис. 2.

Тому семантична модель взаємодії сервісів повинна враховувати вирішення вікід проблем.

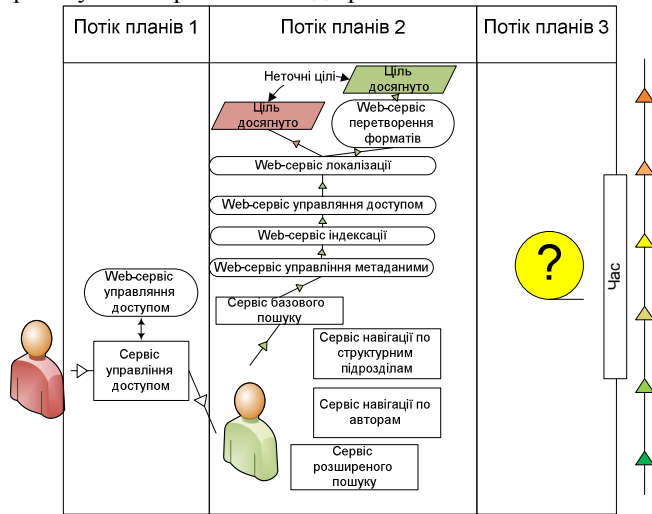


Рис.2 Приклад вікід проблеми

III. ОСОБЛИВОСТІ ПОБУДОВИ СЕМАНТИЧНОГО СЕРЕДОВИЩА

WSMO (WSMO - Web Service Modeling Ontology) підхід в якому відображено всі аспекти які пов'язані з семантичними сервісами, які доступні через веб інтерфейс. Кінцевою ціллю цього підходу являється представлення інформації для автоматичного машинного вирішення задач дослідження, вибору, композиції, співставлення, виконання та моніторингу.

WSMO оперує 4 головними поняттями:

Онтології представляють термінологію яку використовують інші компоненти WSMO. Отологія описує предметну область в якій функціонують веб-сервіси. Онтологія представляє в машино-читаємому форматі термінологію та відношення між елементами термінології.

Веб-сервіси представляють обчислювальні об'єкти, таким чином, що в деякій предметній області ці об'єкти являють собою сервіси. Веб-сервіси включає в себе можливості, інтерфейси, внутрішню роботу все це описано в рамках термінології визначеної онтологіями.

Цілі представляють побажання користувача, відповідно для задоволення яких можна відшукати веб-сервіс.

Посередник – описує ті елементи, які відповідають за вирішення питань функціональної сумісності між іншими елементам WSMO. Саме посередник займає важливу роль оскільки він вирішує задачі несумісності на рівні даних, процесів та протоколів.

Синтаксис WSMO визначається мовою WSML - Web Service Modeling Language.

З урахуванням запропонованої нами парадигми семантичне середовище в рамках WSMO буде мати таку структуру.

Специфікація сервісів складається з специфікації можливостей (Capability) та інтерфейсів (Interface) в WSML.

Функціональність відносно вимог та умов описуються в WSML в формі можливостей (Capability). Можливості вимог є частиною цілі, а можливості умов є частиною веб-сервісу. Стиль взаємодії сторони, яка виконує запит та провайдера описується в інтерфейсах як частина цілі та веб-сервісу відповідно.

Можливості являють собою формальне описання функціональних вимог до веб-сервісів. *Попередні умови* описують вимоги на вході сервісу, *вихідні умови* описують відношення між входом та виходом сервісу, *припущення* описують ті умови які повинні постійно дотримуватися, але не можуть бути перевірені завчасно та *ефекти* які описують реальні результати виконання веб-сервісу, і не є відображеними у вихідних умовах. У цілі чи веб-сервісу може бути тільки одна специфікація можливостей. Слід зазначити для сенсорних сервісів ефекти будуть нульовими.

В WSML ціль може вимагати декілька інтерфейсів і веб-сервіс може надавати декілька інтерфейсів. Хоча специфікація інтерфейсів не є обов'язковою, з точки зору ціле-орієнтованої парадигми визначення інтерфейсів є обов'язковим елементом.

Хореографія інтерфейсу веб-сервісу WSMO, описує поведінку веб-сервісу з точки зору клієнта, такий підхід знаходиться у відповідності до визначення у <http://www.w3.org/TR/ws-gloss/>. Хореографія WSMO керується через механізми станів, і використовує методологію машини Абстрактних станів (Abstract State Machine - ASM) [15]. ASM використовується, щоб абстрактно описати поведінку сервісу, чи поведінку клієнта який взаємодіє з сервісом. ASM вибрана тому, що для задоволення запиту на обслуговування може знадобитися декілька послідовних кроків взаємодії. Ці взаємодії можуть бути описані як кумулятивна зміна параметрів свого стану в процесі виконання машини абстрактних станів.

Розглянемо одиничний сервіс який обслуговує клієнтський запит. Описання інтерфейсу хореографії WSMO базується на правилах які є частиною ASM: основою правил є *стани* системи; *сигнатура* визначає статичну частину опису стану; *базові факти* моделюють динамічну частину опису стану; *транзитивні правила* моделюють зміну стану, змінюючи значення параметрів базових фактів, відповідно до множини імпортованих онтологій.

Для визначення сигнатури використовують отологію WSMO, тобто визначення понять їх атрибутів, відношень і аксіом. Замість динамічної зміни значень функцій, як це представлено в ASMs, ми визначаємо динамічну зміну екземплярів класу їх атрибутів в онтологіях стану системи. Інтерфейс хореографії описує взаємодію відносно одного екземпляру класу хореографії. Ключовим компонентом в порівнянні з ASMs, як було сказано вище, є те що сигнатура визначена в термінах онтології WSMO, а формальною мовою представлення є WSML. Це

призводить до поняття Еволюційної Онтології (по аналогії до Еволюційної Алгебри в ASMs), оскільки хореографія ASM фактично змінює значення понять і відношень в межах онтології.

В рамках нашої моделі транзитивні правила формують цілі композиції веб-сервісів. Оскільки транзитивні правила визначають можливі переходи між станами, загальна семантика цих переходів визначається в цілях. Таким чином цілі в WSMO повинні визначати обмеження можливих переходів станів внаслідок виконання транзитивних правил. Тобто це дасть втілити припущення, що наявна множина сервісів може досягнути деякі наперед невідомі цілі, які можуть бути обрані користувачем. Очевидно, що чим слабше специфікована ціль тим простіше її досягнути. Це дозволить значно покращити функціональність та якість обслуговування кінцевих користувачів електронної бібліотеки, оскільки система сама буде визначати можливі сервіси які будуть надаватися користувачу, а поряд виникає можливість індивідуального обслуговування. В наслідок цього породжується поняття *індивідуальної електронної бібліотеки*, що передбачає не тільки індивідуалізацію контенту але й індивідуалізацію сервісів для роботи з цим контентом. Таким чином користувачу система пропонує можливі операції з DO та об'єднання цих операцій, користувачу залишається лише обрати найбільш прийнятливий для нього варіанти.

Також на відміну від класичних сучасних бібліотечних систем де в основному класифікація DO відбувається на основі метаданих, наша парадигма побудови ЕБ передбачає оперування з семантичними моделями DO, що дасть змогу вирішити низку проблем пов'язаних роботою науковців, це проблеми: навігації по електронним ресурсам; розуміння змісту наукових публікацій; пошуку інформації в предметних областях про які науковець немає чіткого уявлення; отримувати останню достовірну інформацію про досягнення інших науковців; усувати конфлікти між різночасовими версіями наукових робіт; піднести на якісно новий рівень систематизацію знань людства.

ЛІТЕРАТУРА

- [1] J. C. R. Licklider. Libraries of the Future. MIT Press, Cambridge, Massachusetts, 1965.
- [2] D. M. Levy and C. C. Marshall. Going digital: a look at assumptions underlying digital libraries. Communications of the ACM, 38(8):77–84, April 1995.
- [3] C. L. Borgman. What are digital libraries? competing visions. Information Processing and Management, 35(3):227–243, January 1999
- [4] T. Kochtanek and K. K. Hein. Delphi study of digital libraries. Information Processing and Management, 35(3):245–254, 1999.
- [5] IEEE Technical Committee on Digital Libraries <http://www.ieee-tcdl.org>
- [6] Position Statement Past Chairman: Erich J. Neuhold, Fraunhofer-IPSI, Germany Administration Assistant:

Marcello L'Abbate, Fraunhofer-IPSI, Germany

<http://www.ieee-tcdl.org/posstatement.html>

- [7] Yves Petinot, C. Lee Giles, Vivek Bhatnagar, Pradeep B. Teregowda, Hui Han, Isaac Council. A Service-Oriented Architecture for Digital Libraries
- [8] Streams, structures, spaces, scenarios, societies (5s): A formal model for digital libraries (2004) by Marcos André Gonçalves, Edward A. Fox, Layne T. Watson
- [9] Semantic Sensor Web Amit Sheth, Cory Henson, and Satya S. Sahoo • Kno.e.sis Center, Wright State University
- [10] M. Agosti, N. Ferro. A Formal Model of Annotations of Digital Content. ACM Transactions on Information Systems (TOIS), 2008, 26(1):3:1-3:57
- [11] M. Agosti, N. Ferro. A Formal Model of Annotations of Digital Content. ACM Transactions on Information Systems (TOIS), 2008, 26(1):3:1-3:57
- [12] A Service-Oriented Architecture for Digital Libraries Yves Petinot, C. Lee Giles, Vivek Bhatnagar, Pradeep B. Teregowda, Hui Han, Isaac Council
- [13] Oh SG., Lee D., Kumara S.R.T., A comparative Illustration of AI planning-based web services composition, ACM SIGecom Exchanges, vol. 5, Dec. 2005, pp. 1-10.
- [14] Gary Klein, Klein Associates Division of ARA Flexecution as a Paradigm for Replanning, Part 1 SEPTEMBER/OCTOBER 2007 (Vol. 22, No. 5) pp. 79-83 1541-1672/07/\$25.00 © 2007 IEEE Published by the IEEE Computer Society
- [15] Abstract State Machine <http://www.eecs.umich.edu/gasm/>